**DESCRIPTION**

The program detects Morse code raw data input provided from the Phidget 1018 Analog Sensor and converts it into user-readable plain text format.

The program performs two major actions:

1. **Detect the Morse code raw data input from the Phidget 1010 sensor:**
2. The Phidget 1018 Analog sensor captures input from the user.
3. User enters the Morse code by tapping the sensor button on the Phidget.
4. Every key press signifies an input value.
5. On the Phidget, each key press enforces some analog signals that are pressure readings. These readings are measured in voltage units. The sensor registers these forces on a scale 1 to 1000. But, the program does not use these values to determine the plain text message.
6. The input Morse Code interpretation scheme is as follows:
7. Each Morse code value is comprised of dots, dashes and spaces.
8. A combination of dot(s), dash(s) and space(s) represents one raw data value. This value is either an alphabet from a-z or a number from 0-9.
9. For example, Morse Code for ‘a’ is dot-space-dash (. -). Morse code for ‘0’ is dash-space-dash-space-dash-space-dash-space-dash (- - - - -).
10. In the encoding scheme used in the program, a dot is represented by ‘0’, a dash is represented by ‘1’, a space within a character is represented by ‘4’ and a space between two words is represented by a ‘33’.
11. For example, ‘a’ is now ‘041’. ‘0’ is now ‘141414141’. ‘a i’ will be represented as ‘04133040’ as ‘i’ is ‘040’. ‘s o s’ is ‘0404033141413304040’.
12. The above values are stored in a HashMap<String, String>. Here, the key is string representation of the dot-dash-space representation of a Morse code that, is “041”, “040” and so forth.
13. The value corresponding to each key is the alphabet/number representation. If key is “040”’ its corresponding value is “a”, if key is “041414141”’ its corresponding value is “1” and so forth.
14. The raw data message entered by the user is encoded into numbers that represent several key values. For example, “hi” is encoded as “0404040040” where “h” is 04040 and “i” is “40040”. Each substring of the entire raw message string is checked against each key of the HashMap. If a match is found, the key in to the string message is replaced by the corresponding value of the key. Hence, on successful key detection, the message “0404040040” is converted to “hi” and displayed on the screen.
15. Each sensor input value is detected based on time duration. This is the approach for detecting and encoding the raw message input into key values for the HashMap.
16. A ‘dot’ value is detected if the sensor key is pressed for 0-2 seconds. A ‘dash’ value is detected if the sensor key is pressed for 3-4 seconds. A ‘space’ within alphabet/number value is detected if the sensor key is pressed for 5-6 seconds. A ‘space’ between two words is detected if the sensor key is pressed for more than 6 seconds. The duration of each key press enables creation of raw Morse code key.
17. The time for each ley press is measured based on the total time of the key press. The start time and the end time of the key press is noted to determine the total duration. Based on the time duration, appropriate values are appended to a StringBuffer to populate the raw Morse code message.
18. Thus, if there are 5 sensor key presses of 1 second-5 seconds-1 second-5 seconds-1 second then the corresponding Morse code is “04040” which represents an alphabet “s”.
19. Based on the number of inputs and the time per input, the Morse code raw values are detected and eventually encoded to form plain text message which is displayed on the screen as an output.
20. **Encode this raw data into user readable format:**
21. The raw Morse code message string is then encoded using a HashMap iterator and various String operations to determine the plain text message.
22. Each substring of the message is checked against each key of the HashMap. When a match is found, the corresponding value against the key is replaced in the raw message. This match is based on the size of each key starting from matching the key with the highest length.
23. Hence, the keys with length 9 are matched first, then length 8, length, 7, length 5, length 3, length 3 and eventually length 1. This is done to avoid any misrepresentation of the Morse code raw input.
24. Once this encoding is completed, the plain text user readable message is displayed to the user on the screen.
25. The program detects all alphabets, all numbers, in consecutive or different order, detects the message “s o s” and several words like “hi prachi”, “hi al”, “help 786” and so forth.

**SOFTWARE AND TOOLS**

Phidget 1018 Analog sensor, Eclipse Juno.

**IMPLEMENTATION**

1. The HaspMap contains the following key and values:

"041", "a"

"1404040", "b"

"1404140", "c"

"14040", "d"

"0", "e"

"0404140", "f"

"14140", "g"

"0404040", "h"

"040", "i"

"0414141", "j"

"14041", "k"

"0414040", "l"

"141", "m"

"140", "n"

"14141", "o"

"0414140", "p"

"1414041", "q"

"04140", "r"

"04040", "s"

"1", "t"

"04041", "u"

"0404041", "v"

"04141", "w"

"1404041", "x"

"1404141", "y"

"1414040", "z"

"041414141", "1"

"040414141", "2"

"040404141", "3"

"040404041", "4"

"040404040", "5"

"140404040", "6"

"141404040", "7"

"14141040", "8"

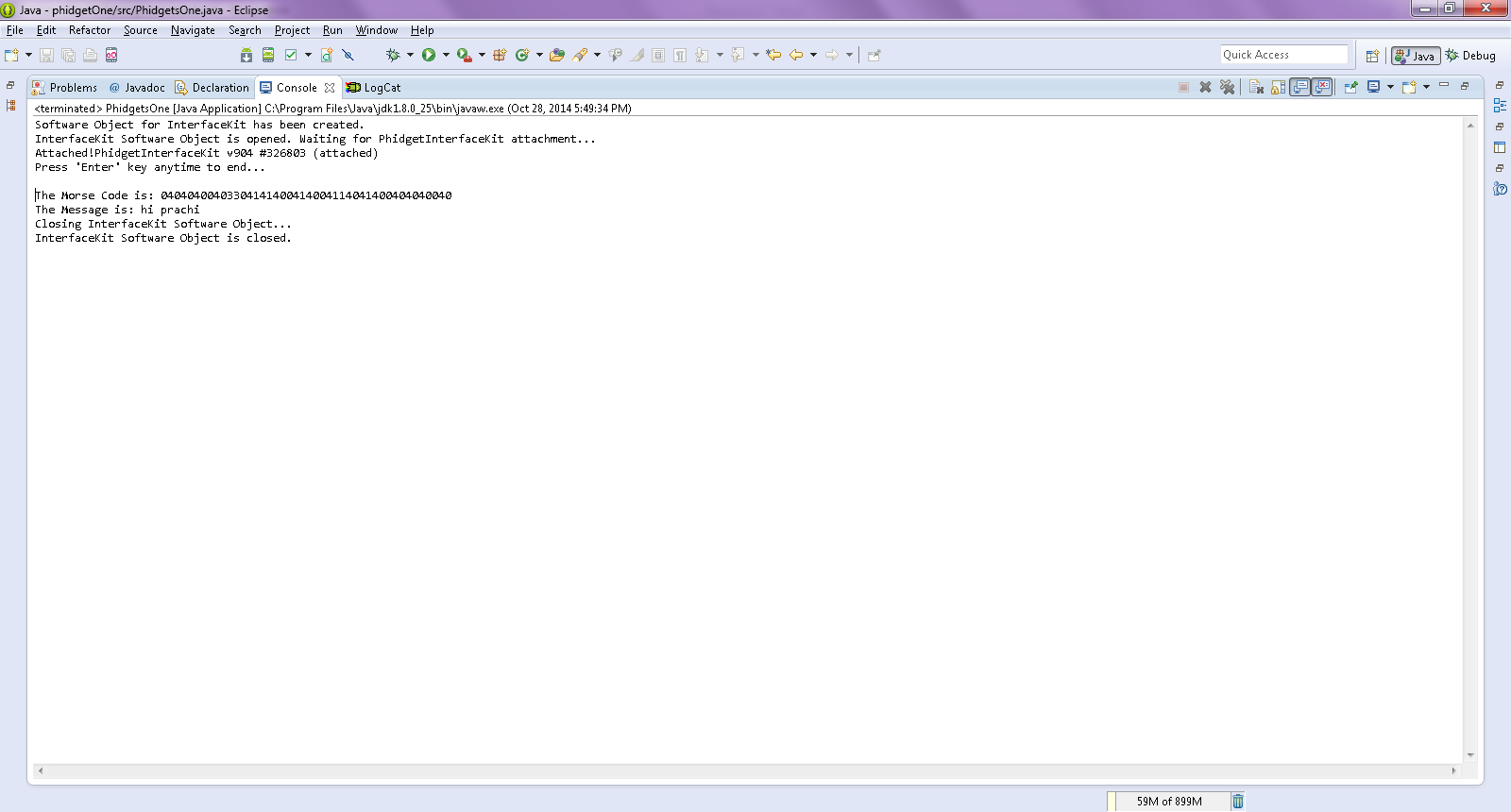
"141414140", "9"

"141414141", "0"

"33", " " [Space within an alphabet/number]

1. Once the program is started, the Phidget is detected.
2. On successful detection, the user is prompted to start entering the Morse code. User presses sensor key for specific duration as per the message he/she wants to send across.
3. A user can press ‘Enter’ key once he/she has completed entering the entire Morse code raw message.
4. On ‘Enter’ key press, the Morse code raw data is encoded and the plain text is displayed on the screen.
5. The InterfaceKit object detects the connected Phidget sensor.
6. The SensorChangeListener event of the InterfaceKit detects any key presses and changes to the sensor input. On sensor input change, the SensorChangeEvent method captures the input based on the time duration. The time is in seconds. On key press, the LED on the Phidget glows.
7. Once the message is displayed on the screen, the Phidget connection is closed by the InterfaceKit.
8. The Phidget21.jar library is used to communicate with the Phidget 1018 sensor.

The plain text message output from a sample raw Morse code input string “**04040400403304141400414004114041400404040040**” gives the output “**hi prachi**” is:



**REFERENCES**

http://www.phidgets.com/docs/1018\_User\_Guide

http://www.phidgets.com/docs/Language\_-\_Java

http://www.phidgets.com/documentation/web/javadoc/

http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html

http://www.tutorialspoint.com/java/java\_hashmap\_class.htm